

Robot Networks with Homonyms: The Case of Patterns Formation

Zohir Bouzid and Anissa Lamani

¹ University Pierre et Marie Curie - Paris 6, LIP6-CNRS 7606, France. zohir.bouzid@lip6.fr

² University of Picardie Jules Verne, Amiens, France. anissa.lamani@lip6.fr

Abstract. In this paper, we consider the problem of formation of a series of geometric patterns [4] by a network of oblivious mobile robots that communicate only through vision. So far, the problem has been studied in models where robots are either assumed to have distinct identifiers or to be completely anonymous. To generalize these results and to better understand how anonymity affects the computational power of robots, we study the problem in a new model, introduced recently in [5], in which n robots may share up to $1 \leq h \leq n$ different identifiers. We present necessary and sufficient conditions, relating symmetricity and homonymy, that makes the problem solvable. We also show that in the case where $h = n$, making the identifiers of robots invisible does not limit their computational power. This contradicts a result of [4]. To present our algorithms, we use a function that computes the Weber point for many regular and symmetric configurations. This function is interesting in its own right, since the problem of finding Weber points has been solved up to now for only few other patterns.

1 Introduction

Robot networks [9] is an area of distributed computing in which the object of the study is the positional (or spacial) communication paradigm [1]: robots are devoid of any means of *direct* communication; instead, they communicate *indirectly* through their movements and the observation of the current positions of their peers. In most of the studies, robots are oblivious, *i.e.* without any memory about their past observations, computations and movements. Hence, as with communication, memory is *indirect* in some sense, it is collective and spacial. Since any algorithm must use a kind of memory, resolving problems in the context of robot networks is the art of making them remember without memory [6]. But this indirect memory has its limits, and one of the main goals of research in this field is to precisely characterize the limiting power of obliviousness.

The problem of *formation of series of patterns* (or patterns formation) is perhaps the best abstraction that captures the need of robots to have a form of memory even when the model does not provide it directly. In this problem, introduced by Das et al. in [4], robots are required to form periodically a sequence of geometric patterns $S = \langle P_1, P_2, \dots, P_m \rangle$. At each instant, robots should be able to know, just by observing the environment, which patterns they are about to form. In [4], the authors study the problem in both anonymous and eponymous systems. In particular, they show that the only formable series when robots are anonymous are those in which all the patterns of the series have the same symmetricity. In contrast, if robots are endowed with identifiers, nearly all possible series are formable.

The gap between the two worlds, one in which almost nothing is possible and another one where everything is possible, illustrates how much anonymity can be a limiting factor when we are interested to computability. This brings us to ask the following question: is there a possible model where robots are neither eponymous nor completely anonymous ? and if so, what are the possible series we are able to form ? In a recent paper [5], Delporte et al. introduce a new model of “partial anonymity” which they call *homonymy* and they apply it to study Byzantine agreement. In this model, the number of distinct identifiers in the system is given by a parameter h which may take any value between 1 and n . In the current paper, we inject this notion of homonymy to the Suzuki-Yamashita model [9, 4] which creates a new model that we call *robot networks with homonyms*.

Studying the problem of patterns formation in this context allows us to get a better insight on how the combined effect of anonymity and obliviousness affects the computational power of robots. We consider series of patterns where all robots are located in *distinct* positions and we assume that identifiers are *invisible*. Our main result is to prove that for a series $S = \langle P_1, P_2, \dots, P_t \rangle$ to

be formable, it is necessary and sufficient that the number of labels h to be strictly greater than $\frac{n}{\text{SNCD}(\text{sym}(P_i), \text{sym}(P_{i+1}))}$ where P_i, P_{i+1} are any two successive patterns of S , $\text{sym}(P)$ denotes the symmetricity of P and $\text{SNCD}(x, y)$ is equal to the smallest divisor of x that does not divide y if any, $n + 1$ otherwise.

To present our algorithms, we use a function that computes the Weber point [10] for many regular configurations, *i.e.* all those in which there is a sort of rotational symmetry around a point. Given a point set P , the Weber point c minimizes $\sum_{r \in P} \text{distance}(x, r)$ over all points x in the plane. Our result may be interesting in its own right, since the problem of finding Weber points has been solved up to now for only few other patterns (e.g. regular polygon [2], a line [3]).

Finally, we consider the case where ($n = h = 3$) and where multiplicity points (in which many robots are located) are allowed. In this setting, we prove that robots are able to form any series of patterns, contradicting a result of [4]. This has an interesting consequence: it means that making the identifiers of robots invisible does not limit their computational power in the considered model, *i.e.* they can form the same series of patterns as robots with visible identifiers.

Roadmap The paper is made up of six sections. In Section 2 we describe the computation model we consider in this paper. Section 3 defines the important notions of symmetricity, regularity and their relations to Weber points. In Section our algorithms for the detection of Weber points are presented. Then, we present in Sections 5 and 6 the necessary and sufficient conditions that makes series of geometric patterns formable in our model. In Section , we address the special case where robots have distinct identifiers and where multiplicity points are allowed. Finally, we conclude this paper in Section 8

2 Model

Our model is based on a variation of the ATOM model [9] used in [4], to which we introduce the notion of homonymy [5]. The system is made up of n mobile robots r_1, \dots, r_n that communicate only through vision. That is, robots are devoid of any mean of direct communication, the only way of them to communicate is by observing the positions of their peers (a “read”) and by moving in the plane (a “write”). To each robot is assigned an identifier (or label) taken from a set of h distinct identifiers $\{1, \dots, h\}$. The parameter $1 \leq h \leq n$ is called the *homonymy* of the system. The identifier of robot r is denoted by $\text{label}(r)$. When two robots have the same label we say that they are *homonymous*. Robots are *oblivious* in the sense that they do not have any memory of their past observations, computations and movements. Hence, their actions are based entirely on their currently observed configuration and their label. Each robot is viewed as a *point* in a plane, thus multiple robots may lie in the same location forming a *multiplicity* point. We consider multiplicity points only in Section 7. There, we assume that robots are able to know exactly how much robots are located in each point. This capability is known as the global strong *multiplicity detection*. We assume also that robots do not obstruct the vision of each others. Robots are *disoriented*, that is, each robot has its own local coordinate system with its own origin, axis, and unit of length which may change at each new activation. However, as in [4], we assume that robots share the same notion of clockwise direction, we say that they have the same *chirality*.

The execution unfolds in *atomic* cycles of three phases Look, Compute and Move. During the Look phase, an activated robot take a snapshot of the environment using its visual sensors. Then it calculates a destination in the Compute phase. The chosen destination is based solely on the label of the robot and the snapshot it obtained in the preceding phase. Finally, In the Move phase, the robot *jumps* to its destination. A subset of robots are chosen for execution (activated) at each cycle by a fictional external entity called a *scheduler*. We require the scheduler to be fair *i.e.* each robot is activated infinitely often. Robots that are activated at the same cycle execute their actions synchronously and atomically.

Notations Given two distinct points in the plane x and y , $|x, y|$ denotes the Euclidean distance between x and y , $[x, y]$ the line segment between them and (x, y) is the line that passes through both of them. Given a third point c , $\angle(x, c, y, \odot)$ denotes the clockwise angle at c from x to y . $\angle(x, c, y, \odot)$ is defined similarly. When the information about orientation can be understood from context, we may omit the parameters \odot and \odot . Note that $\angle(x, c, y, \odot)$ denotes both the angle and its size, the difference between the two notions should be clear from context. Given a configuration P , its smallest enclosing circle, denoted $\text{SEC}(P)$, is the circle of minimum diameter such that every point in P is either on or in the interior of this circle. Given a circle C , $\text{rad}(C)$ and $\text{diam}(C)$ denote its radius and diameter respectively. A point p belongs to C , written $p \in C$ if it is on or inside C . If S is a set, we denote its cardinality by $|S|$.

Problem Definition [4] A configuration C of n robots is a multiset of n elements $\{(p_1, l_1), \dots, (p_n, l_n)\}$ where p_i is the position of robot r_i and l_i is its label. The set of positions $L(C)$ is the set of points occupied by at least one robot in C . A pattern P is represented by a set of n distinct points $\{p_1, \dots, p_n\}$. Two patterns are said isomorphic if they can be obtained from each other by way of translation, rotation and uniform scaling. $\text{size}(P)$ is the cardinality of P . We say that a system of robots has formed the pattern P if the set of points of the current configuration $L(C)$ is isomorphic to P . **The Problem of Formation of Series of Patterns** is defined as follows. As input, robots are given a periodic series of patterns $\langle P_1, P_2, \dots, P_m \rangle^\infty$. It is required that $\forall \tau : \forall P_i \in S : \exists \tau_i : \text{robots form } P_i \text{ at } \tau_i$.

3 Symmetricity, Regularity and Weber Points

In this section we formally define two metrics that quantify how much configurations of *distinct* robots may be symmetric: the strongest one, called symmetricity and the weaker one, called regularity. We then explain how these two notions relate to Weber points. We start by defining a polar coordinate system which we use to state our definitions and algorithms.

3.1 Polar Coordinate System

In the remaining of the paper, we express the locations of robots using a local *polar* coordinate system based on the SEC of the current configuration and the position of the local robot [7]. The *center* c of the coordinate system is common to all robots and it coincides with the center of the SEC. In contrast, the *unit of measure* is local to each robot r and its definition depends on whether the robot is located on c or not. In the first case, the point $(1, 0)$ of the local coordinate system of r is any point in the plane that is not occupied by a robot. In the second case, the point $(1, 0)$ is the current location of r . The positive common *clockwise orientation* is provided by the underlying model. Note that (d, θ) denotes the point located at distance d from c and angle θ .

3.2 Symmetricity

Definition 1 (View). *The view of robot r , denoted $\mathcal{V}(r)$, is the current configuration expressed using the local polar coordinate system of r .*

Notice that since robots are located in distinct positions, our definition of the view implies that if some robot is located at c , then its view is unique, *i.e.* it cannot be equal to the view of another robot.

Now we define the following equivalence relation on robots based on their views.

Definition 2 (\sim). *Given a configuration P , given any two robots $r, r' \in P$:*

$$(r \sim r') \Leftrightarrow (\mathcal{V}(r) = \mathcal{V}(r'))$$

The equivalence class of r is denoted by $[r]$.

Property 1. [4] Let $r \in P$. If $|[r]| > 2$, $[r]$ is a set of robots located at the vertices of a convex regular polygon with $|[r]|$ sides whose center c is the center of $SEC(P)$.

Definition 3 (Symmetricity). The symmetricity of a configuration P , denoted $\text{sym}(P)$, is the cardinality of the smallest equivalence class defined by \sim on P . That is, $\text{sym}(P) = \min\{|[r]| \mid r \in P\}$. If $\text{sym}(P) = m$, we say that P is m -symmetric.

Note that despite the fact that our definition of symmetricity is different from the one used in [4], it is still equivalent to it when configurations does not contain multiplicity points, the case in which we are interested in the current paper.

The next lemma follows from our definition of symmetricity and the fact that the view of a robot located at c is unique.

Lemma 1. If a configuration P of distinct robots is m -symmetric with $m > 1$, then no robot of P is located in c .

In the following lemma we prove that if no robot is located at the center of $SEC(P)$, then all the equivalence classes have the same cardinality.

Lemma 2. Let P be a configuration of n distinct robots, and let c be the center $SEC(P)$. If no robot in P is located at c , it holds that:

$$\forall r, r' \in P : |[r]| = |[r']| = m$$

Proof. We assume that no robot is located in c and we prove the following equivalent claim.

$$\forall r \in P : \forall r' \in P : |[r']| \geq |[r]|.$$

Fix $r \in P$ and let $x = |[r]|$. If $x = 1$, the claim holds trivially. Hence we assume in the following that $x > 1$. Let $[r] = \{r_1, \dots, r_x\}$. The indices define a polar ordering of the robots around c and their addition is done $(\text{mod } n) + 1$. It holds that $r' \in \angle(r_i, c, r_{i+1})$ for some $i \in \{1, \dots, x\}$. But $\mathcal{V}(r_i) = \mathcal{V}(r_{i+1})$ which means that there is a rotational symmetry of angle $\angle(r_i, c, r_{i+1})$ around c . Hence, there exists a robot r'' such that $r'' \in \angle(r_{i+1}, c, r_{i+2})$ and $r'' \sim r'$. By repeating this argument, we find x robots that are equivalent to r' (including itself). Hence $|[r']| \geq x = |[r]|$. This proves the lemma. \square

Lemma 3. Let P be a m -symmetric configuration of n distinct points with center (of symmetricity) c and with $m > 1$. There exists a partition of P into $x = n/m$ subsets S_1, \dots, S_x such that each of them is a convex regular polygon of m sides with center c .

Proof. ($m > 1$) implies that no point of P is located in c (by Lemma 1). Hence, we can apply Lemma 2 and we deduce the existence of $x = n/m$ equivalence classes in $P \sim$ with equal cardinality (m). Denote them by S_1, \dots, S_x . According to Property 1, each of these subsets consists in set of robots located at the vertices of a convex regular polygon of m sides with center c . \square

The following lemma will be used by our algorithm for series formation in Section 6 to break the symmetricity of configurations.

Lemma 4. Let P be a m -symmetric configuration of n distinct points with $m > 1$. If $h > n/m$, there exists two robots r and r' such that $\mathcal{V}(r) = \mathcal{V}(r')$ and $\text{label}(r) \neq \text{label}(r')$.

Proof. Assume not towards contradiction. This implies that all robots that have the same view in P have the same label. Note that since P is m -symmetric, we have exactly n/m equivalence classes (Lemma 2) of equal cardinality. As all robots belonging to the same equivalence class are assumed to have the same label, we have $h = n/m$. But we assumed $h > n/m$, contradiction. \square

3.3 Regularity

In this section we formally define a weaker form of symmetry that we call regularity. In the next section we state its precise relation with symmetricity. We start by giving some useful definitions:

Definition 4 (Successor). *Given a set P of distinct points in the plane, and $c \notin P$ a fixed point. Given some polar ordering of points in P around c . Let us denote by $S(r, c, \curvearrowright)$ the clockwise successor of r according to the clockwise polar ordering of points P around c . The anticlockwise successor of r , denoted $S(r, c, \curvearrowleft)$, is defined similarly.*

Formally [7], $r' = S(r, c, \curvearrowright)$ is a point of P distinct from r such that:

- *If r is not the only point of P that lies in $[c, r]$, we take r' to be the point in $P \cup [c, r]$ that minimizes $|r, r'|$.*
- *Otherwise, we take r' such that no other point of P is inside $\angle(r, c, r')$. If there are many such points, we choose the one that is further from c .*

When the center c and the clockwise orientation are clear from context or meaningless, we simply write $S(r, c)$, $S(r, \curvearrowright)$ or $S(q)$ to refer to the successor of r .

Definition 5 (k -th Successor). *The k -th clockwise successor of r around c , denoted $S^k(r, c, \curvearrowright)$ (or simply $S^k(q)$), is defined recursively as follows:*

- $S^0(r) = r$ and $S^1(r) = S(r)$.
- If $k > 1$, $S^k(r) = S(S^{k-1}(r))$.

Definition 6 (String of angles). *Let P be a set of distinct points in the plane, and $c \notin P$ a fixed point. The (clockwise) string of angles of center c in r , denoted by $SA(r, c, \curvearrowright)$ is the string $\alpha_1 \dots \alpha_n$ such that $\alpha_i = \angle(S^{i-1}(r), c, S^i(r))$. The anticlockwise string of angles, $SA(r, c, \curvearrowleft)$, is defined in a symmetric way. Again, when the information about the center of the string c and/or its clockwise orientation is clear from context, we simply write $SA(r, c)$, $SA(r, \curvearrowright)$ or $SA(r)$.*

Definition 7 (Periodicity of a string). *A string x is k -periodic if it can be written as $x = w^k$ where $1 \leq k \leq n/2$. The greatest k for which x is k -periodic is called the periodicity of x and is denoted by $\text{per}(x)$.*

The following property states that the periodicity of the string of angles does not depend on the process in which it is started nor on the clockwise or anticlockwise orientation.

Lemma 5. *Let P be a set of distinct points in the plane, and let $c \notin P$ be a point in the plane. The following property holds:*

$$\exists m : \forall r \in P : \forall d \in \{\curvearrowright, \curvearrowleft\} : \text{per}(SA(r, c, d)) = m$$

Proof. Follows from the fact that a periodicity of a string is invariant under symmetry and rotation. \square

Theorem 1. *Given is a set of points P , a center $c \notin P$. There exists an algorithm with running time $O(n \log n)$ that computes $SA(c)$.*

Proof. Fix some $r \in P$. We show how to compute $SA(r, c, \curvearrowright)$. The other cases are similar. We use an array $T[n]$ of n cells. As a first step, we compute for each $p_i \in P$ the angle $\angle(r, c, p_i, \curvearrowright)$ and put the result in $T[i]$. Note that this step takes $O(n)$ time. Then we sort T in increasing order ($O(n \log n)$ steps). $SA(r, c, \curvearrowright)$ is the string $\alpha_1 \dots \alpha_n$ such that $\forall i \in [1, n-1] : \alpha_i = T[i+1] - T[i]$ and $\alpha_n = 2\pi - T[n]$. The whole algorithm runs in $O(n \log n)$ time. \square

The lemma means that when it comes to periodicity, the important information about a string of angles is only its center c . Hence, in the following we may refer to it by writing $SA(c)$.

Definition 8 (Regularity). Let P be a set of n distinct points in the plane. P is m -regular (or regular) if there exists a point $c \notin P$ such that $\text{per}(SA(c)) = m > 1$. In this case, the regularity of P , denoted $\text{reg}(P)$, is equal to $\text{per}(SA(c))$. Otherwise, it is equal to 1. The point c is called the center of regularity.

Theorem 2. Given is a set of points P , a center $c \notin P$. There exists an algorithm with running time $O(n \log n)$ that detects if c is a center of regularity for P .

Proof. The algorithm computes $SA(c)$. As proved in Theorem 1, this can be done in $O(n \log n)$ time. Then it computes the periodicity of the string in linear time [8]. If the obtained periodicity is greater than 1, it outputs YES, otherwise it outputs NO. \square

3.4 Weber Points

In this section we state some relations between symmetricity and regularity. Then we prove that their centers are necessarily Weber points, hence unique when the configuration is not linear. The following lemma is trivial, its proof is left to the reader. It states the fact that if a configuration is m -symmetric, it is necessarily m -regular.

Lemma 6. Let P be any configuration with $\text{sym}(P) > 1$. It holds that $\text{reg}(P) = \text{sym}(P)$ and the center of regularity of P coincides with its center of symmetricity.

The next lemma shows in what way the regularity of a configuration can be strengthened to become a symmetricity.

Lemma 7. Let P be any configuration with $\text{reg}(P) = m > 1$, and let c be its center of regularity. There exists a configuration P' that can be obtained from P by making robots move along their radius such that $\text{sym}(P') = m$. Moreover, c is the center of symmetricity of P' .

Proof. Let d be the greatest distance between any point of P and c . Note that since $m > 1$, no robot is located in c . We construct a configuration P' by making each point r in P move along its radius towards the point $(d, 0)$ (expressed in its local polar coordinate system). In configuration P' , all the points are at the same distance from c , hence the symmetricity in this case depends only on angles which makes it equal to regularity. Indeed, the equivalence class of each robot r is formed by the set of robots $\{r, S^{n/m}(r, c), \dots, S^{(m-1)n/m}(r, c)\}$. Clearly, $|[r]| = m$. Since r is arbitrary, all the equivalence classes have a cardinality of m . Hence, $\text{sym}(P') = m$ and c is the center of symmetricity of P' . \square

The following lemma states that the center of symmetricity of any configuration P if any is also its Weber point. The same claim was made in [2] in the cases when $\text{sym}(P) = n$ (equiangular) and $\text{sym}(P) = n/2$ (biangular). Our proof uses the same reasoning as theirs.

Lemma 8. Let P be a configuration such that $\text{sym}(P) > 1$ and let c be its center of symmetricity. It holds that c is the Weber point for P .

Proof. Let $m = \text{sym}(P)$. Assume towards contradiction that the Weber point of P is $c' \neq c$. Since P is m -symmetric, it holds that all there $m - 1$ other points that are rotational symmetric to c' around c with angles of $\frac{2\pi}{m}, \dots, \frac{2(m-1)\pi}{m}$. By symmetricity, all these points are also Weber points. But since $m > 1$, this means that the Weber point is not unique. But Weber points are unique when not all the points are located on the same line. Hence c , the only point that is "unique" in P , is the Weber point. \square

Lemma 9. Let P be a configuration such that $\text{reg}(P) = m > 1$ and let c be its center of regularity. It holds that c is the Weber point for P .

Proof. Our argument is similar to [2]. According to Lemma 7, there exists a configuration P' that can be obtained from P by making robots move along their radius. Moreover, $\text{sym}(P') = m$ and c is its center of symmetry. According to Lemma 8, c is the Weber point of P' . But it is known in geometry that a Weber point remains invariant under straight movements of any of the points towards or away from it. Hence, since c is the Weber point of P' , it must be the Weber point of P also. This proves the lemma. \square

The next corollary follows directly from Lemma 9 and the fact that Weber points are unique.

Corollary 1 (Unicity of c). *The center of regularity is unique if it exists.*

4 Detection of Regular Configurations

In this section we show how to identify geometric configurations that are regular. We present two algorithms that detects whether a configuration P of n points given in input is m -regular for some $m > 1$, and if so, they output its center of regularity. The first one detects the regularity only if m is even, it is very simple and runs in $O(n \log n)$ time. The second one can detect any regular configuration, provided that $m \geq 3$. It is a little more involved and runs in $O(n^4 \log n)$ time. We assume in this section that P is not a configuration in which all the points are collinear.

4.1 Preliminaries

In this section, we state some technical lemmas that help us in the presentation and the proofs.

Lemma 10. *Let P be a regular configuration of n distinct points and let c be its center of regularity. Let $m = \text{reg}(P)$. The following property holds:*

$$\forall r \in P : (\angle(r, c, S^{n/m}(r, \odot), \odot) = 2\pi/m) \wedge (\angle(r, c, S^{n/m}(r, \odot), \odot) = 2\pi/m)$$

Proof. Fix $r \in P$. Since P is m -regular, then m is necessarily a divisor of n . Let $x = n/m$. We divide the proof into two parts:

1. Part 1: $\angle(r, c, S^x(r, \odot), \odot) = 2\pi/m$

Let $S = SA(r, c, \odot)$. Assume that $S = \alpha_1 \dots \alpha_n$. Clearly we have that

$$\alpha_1 + \alpha_2 + \dots + \alpha_n = 2\pi$$

Since $n = m \cdot x$, the above sum can be rewritten as follows:

$$(\alpha_1 + \dots + \alpha_x) + (\alpha_{x+1} + \dots + \alpha_{2x}) + \dots + (\alpha_{(m-1)x+1} + \dots + \alpha_{mx}) = 2\pi$$

But S is m -regular. Hence, $\forall i \in [1, n] : \alpha_i = \alpha_{(i+x) \bmod (n+1)}$. This implies that:

$$m \cdot (\alpha_1 + \dots + \alpha_x) = 2\pi$$

But $\angle(r, c, S^x(r, \odot), \odot) = (\alpha_1 + \dots + \alpha_x)$. Hence $\angle(r, c, S^x(r, \odot), \odot) = 2\pi/m$. This finishes the first part of the proof.

2. Part 2: $\angle(r, c, S^x(r, \odot), \odot) = 2\pi/m$

Let $r' = S^x(r, \odot)$ and notice that $S^x(r', \odot) = r$. It holds that

$$\angle(r, c, S^x(r, \odot), \odot) = \angle(r, c, r', \odot) = \angle(r', c, r, \odot)$$

But since $S^x(r', \odot) = r$, we have: $\angle(r', c, r, \odot) = \angle(r', c, S^x(r', \odot), \odot) = 2\pi/m$ (as shown in Part 1). Hence $\angle(r, c, S^x(r, \odot), \odot) = 2\pi/m$.

□

The following lemma proves that when a configuration P is m -regular with m even, then for each point in $r \in P$ there exists a corresponding point ($S^{n/2}(r)$) that lies on the line that passes through r and c with c being the center of regularity of P .

Lemma 11. *Let P be a regular configuration of n distinct points and let c be its center of regularity. The following property holds:*

$$(reg(P) \text{ is even}) \Rightarrow (\forall r \in P : \angle(r, c, S^{n/2}(r)) = \Pi)$$

Proof. Let $m = reg(P)$. By assumption m is even, hence there exists some $m' \in \mathbb{N}^+$ such that $m = 2m'$. Then, it holds that $n/2 = m'(n/m)$. This implies that

$$\angle(r, c, S^{n/2}(r)) = \angle(r, c, S^{n/m}(r)) + \angle(S^{n/m}(r), c, S^{2n/m}(r)) + \dots + \angle(S^{(m'-1)n/m}(r), c, S^{n/2}(r))$$

But since P is m -regular, we have according to Lemma 10 that:

$$\angle(r, c, S^{n/2}(r)) = 2\Pi/m + \dots + 2\Pi/m = m'(2\Pi/m)$$

By replacing $m = 2m'$ we get: $\angle(r, c, S^{n/2}(r)) = \Pi/2$ which proves the claim. □

4.2 Detection of Even Regularity

In this section we present our algorithm for detecting regular configurations when the regularity is even. Note that since regularity is a divisor of n , it holds that n must be also even in this case. It is inspired from Algorithm 2 in [2] and runs in $O(n \log n)$ steps. It is based on the notion of median lines taken from [2].

Definition 9 (*median(r)*). *Given a configuration P of n robots, n even, given r any robot in P , the median line of r , denoted $median(r)$, is the line that passes through r and some other robot r' in P and divides the set of points into two subsets of equal cardinality $\frac{n}{2} - 1$. Note that in our case, contrary to [2], we allow other robots than r and r' to lie in the median. This is explained by the fact that, contrary to them, we allow angles equal to 0.*

Lemma 12. *Let P be a regular configuration of n distinct points with $reg(P) = m$ even. Let c be the center of regularity. The following property holds:*

$$\forall r \in P : median(r) \text{ passes through } S^{n/2}(r) \text{ and } c$$

Proof. According to Lemma 11, $\angle(r, c, S^{n/2}(r)) = \pi$. Hence the lemma follows. □

As a corollary we have that the center of regularity lies in the intersection of all medians.

Lemma 13. *Let P be a regular configuration of n distinct points with $reg(P) = m$ even. Let c be the center of regularity. It holds that*

$$\forall r \in P : median(r) \cap \{c\} = \{c\}$$

Theorem 3. *Given P a configuration of n distinct points with n even. There exists an algorithm running in $O(n \log n)$ steps that detects if P is m -regular with m even, and if so, it outputs m and the center of regularity.*

Proof. The algorithm is described formally in Figure 1. It first computes the convex hull CH of the configuration which takes $0(n \log n)$ operations. Then it chooses an arbitrary point r in CH and computes its $median(r)$. After that, it picks another point r' that belongs to CH but not to $median(r)$. The point r' is well defined since we assume that the points of P are not all collinear. This choice of r' guarantees that $median(r) \neq median(r')$. If the configuration is regular, its center of regularity necessarily lies in the intersection c of the two medians (Lemma 13). Then, it can be checked in $0(n \log n)$ steps whether c is a center of regularity (Theorem 2).

```

(1)  $CH \leftarrow \text{CONVEX HULL}(P)$ 
(2)  $r \leftarrow$  an arbitrary point on  $CH$ 
(3)  $r' \leftarrow$  a point on  $CH$  but not on  $median(r)$ 
(4) If  $((median(r) \cap median(r')) \neq \perp)$ 
(5)    $c \leftarrow median(r) \cap median(r')$ 
(6)    $SA \leftarrow$  string of angles of  $P$  around  $c$ 
(7)    $m \leftarrow per(SA)$ 
(8)   If  $(m > 1)$  RETURN (REGULARITY  $m$  WITH CENTER  $c$ )
(9) Endif
(10) return (NOT REGULAR)

```

Fig. 1. Detection of Even Regularity

□

4.3 Detection of Odd Regularity

Definition 10 (α -Circle).

Given two distinct points x and y and an angle $0 < \alpha < \Pi$, we say that the circle C_{xy} is a α -circle for p and q if $x, y \in C_{pq}$ and there exists a point $p \in C_{xy}$ such that $\angle(x, p, y) = \alpha$.

In the following we present three known properties [2] about α -circles.

Property 2. it holds that $\angle(x, p', y) = \alpha$ for every point $p' \in C_{pq}$ on the same arc as p .

Property 3. If $\alpha = \Pi/2$, the α -circle is *unique* and is called the Thales circle.

Property 4. If $\alpha \neq \Pi/2$, there are *exactly* two α -circles. We denote them in the following by C_{xy} and C'_{xy} . $\alpha = \Pi/2$ can be seen as a special case in which $C_{xy} = C'_{xy}$.

Definition 11 ($C_{xy} \cap C_{yz}$). Given C_{xy} and C_{yz} , we define their intersection, denoted $C_{xy} \cap C_{yz}$, as the point p such that $(p \neq y) \wedge (p \in C_{xy}) \wedge (p \in C_{yz})$. If p does not exist we write $C_{xy} \cap C_{yz} = \emptyset$.

Lemma 14. Let $m \geq 3$. Let P be an m -regular configuration of n distinct points with center $c \notin P$. Let x be any point in P , and let us denote by y and z the points $S^{n/m}(x, c, \odot)$ and $S^{n/m}(x, c, \ominus)$ respectively. Let $C_{xy}, C'_{xy}, C_{xz}, C'_{xz}$ be the $2\Pi/m$ -circles for the appropriate points. It holds that $c \in (C_{xy} \cap C_{xz}) \cup (C'_{xy} \cap C_{xz}) \cup (C_{xy} \cap C'_{xz}) \cup (C'_{xy} \cap C'_{xz})$.

Proof. According to Theorem 10, since P is m -regular, then $\angle(x, c, y, \odot) = 2\Pi/m$ where c is the center of regularity. Hence, either $c \in C_{xy}$ or $c \in C'_{xy}$ (According to Definition 10 and Property 4). Using the same argument, we can also show that either $c \in C_{xz}$ or $c \in C'_{xz}$. Hence:

$$((c \in C_{xy}) \vee (c \in C'_{xy})) \wedge ((c \in C_{xz}) \vee (c \in C'_{xz}))$$

Distributing \wedge over \vee gives us:

$$((c \in C_{xy}) \wedge (c \in C_{xz})) \vee ((c \in C_{xy}) \wedge (c \in C'_{xz})) \vee ((c \in C'_{xy}) \wedge (c \in C_{xz})) \vee ((c \in C'_{xy}) \wedge (c \in C'_{xz}))$$

This proves the lemma. □

Lemma 15. *Given $3 \leq m \leq n$, given P a configuration of n distinct points. Let x be any point in P . The following property holds:*

$$(P \text{ is } m\text{-regular with center } c)$$

$$\Leftrightarrow$$

$$(\exists y, z \in P : (x \neq y \neq z) \wedge (c \in (C_{xy} \cap C_{xz}) \cup (C'_{xy} \cap C_{xz}) \cup (C_{xy} \cap C'_{xz}) \cup (C'_{xy} \cap C'_{xz})))$$

Where $C_{xy}, C_{xz}, C'_{xy}, C'_{xz}$ are $2\pi/m$ -circles of the corresponding points.

Proof. The lemma follows from Lemma 14 by setting $y = S^{n/m}(x, c, \circlearrowleft)$ and $z = S^{n/m}(x, c, \circlearrowright)$. \square

Theorem 4. *Given $3 \leq m \leq n$, given P a configuration of n distinct points. There exists an algorithm running in $O(n^3 \log n)$ that detects if P is m -regular, and if so, it outputs the center of regularity.*

Proof. The algorithm is the following. We fix any robot $x \in P$. Then, for every $y \in P \setminus \{x\}$, for every $z \in P \setminus \{x, y\}$, for every $c \in (C_{xy} \cap C_{xz}) \cup (C'_{xy} \cap C_{xz}) \cup (C_{xy} \cap C'_{xz}) \cup (C'_{xy} \cap C'_{xz})$, we test if c is a center of regularity (Theorem 2, $O(n \log n)$ time). Lemma 4.3 guarantees that if P is m -regular, the test will be conclusive for at least one pair (y, z) of robots. The whole algorithm executes in $O(n^3 \log n)$: we browse all the possible pairs (y, z) , and for each pair we generate up to four candidates for the center of regularity, hence we have $O(n^2)$ candidates. Then, $O(n \log n)$ time is needed to test each candidate. Note that our algorithm follows the same patterns as those presented in [2]: generating a restricted set of candidates (points) and testing whether each of them is a center of regularity. \square

Theorem 5. *Given P a configuration of n distinct points. There exists an algorithm running in $O(n^4 \log n)$ steps that detects if P is m -regular with $m \geq 3$, and if so, it outputs m and the center of regularity.*

Proof. It suffices to generate all the divisors m of n that are greater than 2. Then, for each m , we test if P is m -regular using the algorithm of Theorem 4). When the test is conclusive, this algorithm returns the center of regularity c , so we can output REGULARITY m , CENTER c . If test was inconclusive for every generated m , we output NOT REGULAR. \square

Theorem 6. *Given P a configuration of n distinct points. There exists an algorithm running in $O(n^4 \log n)$ steps that detects if P is m -regular with $m \geq 2$, and if so, it outputs m and the center of regularity.*

Proof. We combine the algorithms of Theorems 3 and 5. First, we test if P is m -regular for some even m using the algorithm of Theorem 3 ($O(n \log n)$). If so, we output m and the center of regularity c which are provided by the called algorithm. Otherwise, we test odd regularity using the algorithm of Theorem 5 ($O(n^4 \log n)$) but by restricting the analysis to only the odd divisors of n (the even divisors were already tested). \square

5 Formation of a Series of Geometric Patterns: Lower Bound

In this section we prove a necessary condition that geometric series have to satisfy in order to be formable. The condition relates three parameters: the number of robots in the system n , its homonymy h and the symmetricity of the patterns to form. It is stated in Lemma 18.

Property 5. [4] For any configuration P of n distinct robots, $\text{sym}(P)$ divides n .

Lemma 16. *Let P be a configuration of n distinct robots with symmetricity s , i.e. $\text{sym}(P) = s$. For any divisor d of s , if $h \leq \frac{n}{d}$, then for any pattern formation algorithm, there exists an execution where all subsequent configurations P' satisfy $\text{sym}(P') = k \cdot d$, $k > 1$.*

Proof. The lemma holds trivially if $s = 1$, hence we assume in the following that $s > 1$. According to Lemma 3, there exists a partition of P into $x = n/s$ subsets S_1, \dots, S_x such that the s robots in each S_i occupy the vertices of a regular convex polygon of s sides whose center is c .

Now, partition each set S_i into $\frac{s}{d}$ subsets $T_{i1}, \dots, T_{i(\frac{s}{d})}$ with $|T_{ij}| = d$ for each $i \in \{1, \dots, x\}, j \in \{1, \dots, \frac{s}{d}\}$. Each subset T_{ij} is chosen in such a way that the d robots belonging to it are located in the vertices of a regular convex polygon of d sides with center c . This choice is possible because d is a divisor of $s = |S_i|$. For example, let r_1, \dots, r_s be the robots of S_i ordered according to some polar ordering around c . T_{i1} is the set of robots $\{r_1, r_{(\frac{s}{d}+1)}, r_{(\frac{2s}{d}+1)}, \dots, r_{(\frac{(d-1)s}{d}+1)}\}$. Clearly, this subset defines a regular polygon of d sides with center c .

There are total of $\frac{s \cdot x}{d} = \frac{n}{d}$ subsets T_{ij} . So we have also a total of $\frac{n}{d}$ concentric regular polygons of d sides. What is important to notice now is that the robots in each T_{ij} have the same view.

Since $h \leq \frac{n}{d}$, there exists a set of labels $|\mathcal{L}| = h$, and a labeling of robots in P such that (1) The same label is assigned to the robots that belong to the same subset T_{ij} . (2) For each label $l \in \mathcal{L}$, there exists a robot $r_i \in P$ such that l is the label of r_i .

Since we assume that algorithms are deterministic, the actions taken by robots at each activation depend solely on they observed view and their identity (label). Hence, two robots having the same view and the same label will take the same actions if they are activated simultaneously. Therefore, the adversary can guarantee that the network will always have a symmetricity $\geq d$ by activating each time the robots that belong to the same T_{ij} together. This way, we are guaranteed to have all the subsequent configurations that consists of a set of $\frac{n}{d}$ concentric regular polygons of d sides. That is, all subsequent configuration have a symmetricity that is a multiple of d . This proves the lemma. \square

The following lemma states a necessary condition for a geometric figure P_j to be formable starting from P_i .

Lemma 17. *If the current configuration P_i has symmetricity $\text{sym}(P_i) = s$, the configuration P_j with symmetricity $\text{sym}(P_j) = s'$ is formable only if (1) $\text{size}(P_j) = \text{size}(P_i)$ and (2) $h > \frac{n}{\text{SNCD}(s, s')}$ where SNCD (read smallest non common divisor) is equal to the smallest x that divides s but not s' if any, $n + 1$ otherwise.*

Proof. Assume towards contradiction that (1) $h \leq \frac{n}{\text{SNCD}(s, s')}$ and (2) P_j is formable. Note that since $h \geq 1$, (1) implies that $\text{SNCD}(s, s') \neq n + 1$. Otherwise we would have $h \leq 0$, contradiction. By definition, $\text{SNCD}(s, s') \neq n + 1$ implies that $\text{SNCD}(s, s') = t$ divides s but not s' .

According to Lemma 16, for any algorithm, there exists an execution starting from P_i where all subsequent configurations P' satisfy $\text{sym}(P') = k \cdot t$, $k > 1$. But $\text{sym}(P_j) = s'$ is not multiple of t , hence P_j is never reached in this execution. This means that P_j is not formable starting from P_i , which contradicts (2). Hence, the lemma is proved. \square

Now, we are ready to state the necessary condition for formation of geometric series. It relates the symmetricity of its constituent patterns and the homonymy of the system.

Lemma 18. *A cyclic series of distinct patterns $\langle P_1, P_2, \dots, P_m \rangle^\infty$ each of size n is formable only if*

$$\forall i \in \{1, \dots, m\} : h > \frac{n}{\text{SNCD}(\text{sym}(P_i), \text{sym}(P_{i+1}))}$$

Proof. Follows from Lemma 18. \square

6 Formation of Series of Patterns: Upper Bound

In this section, we present an algorithm that allows robot to form a series of patterns, provided that some conditions about homonymy and symmetricity are satisfied. The result is stated in the following theorem:

Theorem 7. A cyclic series of distinct patterns $\langle P_1, P_2, \dots, P_m \rangle^\infty$ each of size n is formable if and only if

$$\forall i \in \{1, \dots, m\} : h > \frac{n}{\text{SNCD}(\text{sym}(P_i), \text{sym}(P_{i+1}))}$$

The only if part was proved in Section 5. The remaining of the current section is devoted to the proof of the if part of the theorem.

6.1 Intermediate Configurations

During the formation of a pattern P_i (starting from P_{i-1}), the network may go through several intermediate configurations. We define in the following four classes of intermediate patterns $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and \mathcal{D} . Each one of them encapsulate some information that allows robots to unambiguously determine which pattern the network is about to form. This information is provided by a function, STRETCH , which we define separately for each intermediate pattern.

Definition 12 (Configuration of type \mathcal{A}). A configuration P of n points is of type \mathcal{A} (called *BCC* in [4]) if the two following conditions are satisfied (refer to Figure 6.1):

1. there exists a point $x \in P$ such that the diameter of $\text{SEC}_1 = \text{SEC}(P)$ is a least ten times the diameter of $\text{SEC}_2 = \text{SEC}(P \setminus \{x\})$.
2. SEC_1 and SEC_2 intersect at exactly one point called the base-point (BP).

The point x is called the pivot whereas the point on SEC_2 directly opposite BP is called the frontier point (FP).

$\text{STRETCH}(P)$ is equal to $\lfloor \frac{\text{rad}(\text{SEC}_1)}{(h+1) \cdot \text{rad}(\text{SEC}_2)} \rfloor$.

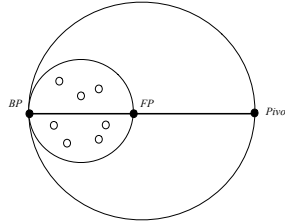


Fig. 2. Figure of type \mathcal{A} .

Definition 13 (Configuration of type $\mathcal{B}(m)$). A configuration P of n distinct robots is of type $\mathcal{B}(m)$ (adapted from $\text{SCC}[m]$ in [4]) with $1 < m < n$, if the following conditions are satisfied:

1. $\text{SEC}_1 = \text{SEC}(P)$ has exactly m points on its circumference which form a regular convex polygon with m sides.
2. Let SEC_2 be the SEC of the robots that are not on the SEC_1 , i.e. $\text{SEC}_2 = \text{SEC}(P \setminus \{r \in P \mid r \text{ is on } \text{SEC}_1\})$. SEC_1 and SEC_2 are concentric such that $\text{rad}(\text{SEC}_1) > 10 \cdot \text{rad}(\text{SEC}_2)$.

$\text{STRETCH}(P)$ is equal to $\frac{1}{2} \cdot \lfloor \frac{\text{rad}(\text{SEC}_1)}{(h+1) \cdot \text{rad}(\text{SEC}_2)} \rfloor$. It can be easily checked that a given configuration cannot be of both types \mathcal{A} and \mathcal{B} (their respective SEC_2 do not intersect).

Definition 14 (Configuration of type $\mathcal{C}(m)$). A configuration P of n distinct robots is of type $\mathcal{C}(m)$ with $1 < m < n$ if 1) it is not of type \mathcal{B} and 2) it is m -symmetric (ref. Definition 3).

$\text{STRETCH}(P)$ is computed as follows. Since P is m -symmetric with $m > 1$, there exists a partition of P into $x = n/m$ subsets S_1, \dots, S_x such that each of them is a convex regular polygon of m sides with center c (Lemma 3). This means that each S_i defines a circle with center c . Assume w.l.o.g that $\forall i \in \{1 \dots x-1\} : \text{rad}(S_i) \leq \text{rad}(S_{i+1})$. $\text{STRETCH}(P) = \text{Max}\{\lfloor \frac{\text{rad}(S_{i+1})}{(h+1) \cdot \text{rad}(S_i)} \rfloor \mid i \in \{1 \dots x-1\}\}$.

Clearly, a configuration of type \mathcal{C} cannot be of type \mathcal{A} since the former is symmetric while the latter is not.

Definition 15 (Configuration of type $\mathcal{D}(m)$). A configuration P of n distinct robots is of type $\mathcal{D}(m)$ with $1 < m < n$ if (1) Points are not all on the same line, (2) P is not of type \mathcal{B} and (3) P is m -regular but not symmetric ($\text{Sym}(P) = 1$).

$\text{STRETCH}(P)$ is computed as follows. Let c be the center of regularity. c can be computed in polynomial time using the algorithm of Section 4 (Theorem 6). Then for each $r_i \in P$, we compute $|c, r_i|$, its distance from c . Assume w.l.o.g. that $\forall i \in \{1 \dots x-1\} : |c, r_i| \leq |c, r_{i+1}|$. $\text{STRETCH}(P) = \text{Max}\{\lfloor \frac{|c, r_{i+1}|}{(h+1) \cdot |c, r_i|} \rfloor \mid i \in \{1 \dots x-1\}\}$. Note that the stretch of configurations of type $\mathcal{C}(m)$ is a particular case of that of type $\mathcal{D}(m)$, but since the former configurations are symmetric, we can compute their stretch without resorting to the computation of the center of regularity.

Decoding the stretch Let F be a one-to-one function [4] that maps each pattern P_i to a real number $t_i = F(P_i)$. If there is a pattern P_i that is of type $\mathcal{A}, \mathcal{B}, \mathcal{C}$ or \mathcal{D} , we exclude the value $\text{STRETCH}(P_i)$ from the domain of F . To simplify the proofs, we assume that $F(P_i) > 10$ for any P_i . When robots are about to form the pattern P_i , they use intermediate configurations with stretch t_i . By computing the stretch, robots can unambiguously identify which configuration they are about to form ($F^{-1}(t_i)$).

6.2 Transitions between Configurations

In this section we describe some algorithms that describe some transformations between patterns.

Lemma 19. Starting from any configuration of type \mathcal{D} with stretch t , there exists an algorithm that builds a configuration of type \mathcal{A} with the same stretch.

Proof. Let P be the initial configuration. Since P is of type \mathcal{D} , it holds that $\text{sym}(P) = 1$. Hence we can elect a leader, i.e. a robot whose view is unique, let it be r . The algorithm work by making r move towards the pivot position of the target \mathcal{A} configuration with appropriate stretch. This is done as follows. First, r computes the stretch $t = \text{STRETCH}(P)$. Then it consider the set of points $Q = P \setminus \{r\}$ and compute their SEC and its center c . Here, we distinguish between two cases:

1. There exists two robots r_2 and r_3 of SEC that are directly opposite of each others. In this case, r_1 jumps to the pivot location that makes r_2 and r_3 occupy the frontier-point and the base-point respectively. Formally, r_1 jumps to a point x located at the line (r_3, r_2) such that r_2 lies between r_3 and x with $|c, x| = (t \cdot (h+1)) \cdot |c, r_2|$. Let P' the obtained configuration. Clearly, P' is of type \mathcal{A} and $\text{STRETCH}(P') = t$.
2. Otherwise, r_1 chooses any point r_2 on SEC and jumps to the pivot position with appropriate stretch such that r_2 becomes the base point. Then, any robot distinct from r_1 and r_2 will jump to the frontier point.

In both cases, we obtain a configuration of type \mathcal{A} with the required stretch. □

Lemma 20. Starting from any configuration of type P_i with stretch $\text{sym}(P_i) = 1$, there exists an algorithm that builds a configuration of type \mathcal{A} with stretch $F(P_{i+1})$.

Proof. The proof is similar to that of Lemma 19, by replacing t with $F(P_{i+1})$. □

The following two lemmas are from [4].

Lemma 21. Starting from any configuration of type \mathcal{A} , it is possible to form any single pattern.

Lemma 22. *Starting from any configuration of type $B(m)$, it is possible to form any single pattern P such that $\text{sym}(P) = k \cdot m, k > 1$.*

Lemma 23. *Consider a robot network of n robots in configuration P_i . Let $\text{sym}(P_i) = m > 1$ and $\text{sym}(P_{i+1}) = m'$. If $h > \frac{n}{\text{SNCD}(m, m')}$, there exists an algorithm that brings the network to a configuration Q such that either Q is of type $\mathcal{B}(x), x < \text{SNCD}(m, m')$ or of type \mathcal{A} both with stretch $F(P_{i+1})$.*

The remaining of this section is devoted to the proof of this lemma.

Function:
 $\text{targetSym}(k)$: The target symmetricity when robots try to form P_{k+1}
 It is equal to the greatest divisor of $\text{sym}(P_{k+1})$ that is $< \text{SNCD}(\text{sym}(P_k), \text{sym}(P_{k+1}))$

Actions:

- (1) $P \leftarrow \text{Observed Configuration}$
- (2) $\text{SEC} \leftarrow \text{SMALLEST ENCLOSING CIRCLE}(P)$
- (3) $c \leftarrow \text{CENTER}(\text{SEC})$
- (4) $\text{sym} \leftarrow \text{sym}(P)$
- (5) **If** $(P = P_i)$ **then** $t \leftarrow F(P_{i+1})$
- (6) **else** $t \leftarrow \text{stretch}(P)$ **endif**
- (7) $\text{id} \leftarrow \text{My Identifier}$
- (8) $\text{rad} \leftarrow \text{RADIUS}(\text{SEC})$
- (9) **if** $(\text{sym} > 1) \wedge ((\text{sym} > \text{targetSym}(k)) \vee (P \text{ is not of type } \mathcal{B}(\text{sym})))$
- (10) $d \leftarrow \text{Min}(|r_i, c|; r_i \in P)$
- (11) $S \leftarrow \{r_i \in P \mid |r_i, c| = d\}$
- (12) $\text{minView} \leftarrow \text{Min}(V(r_i); r_i \in S)$
- (13) $\text{Elected} \leftarrow \{r_i \in S \mid V(r_i) = \text{minView}\}$
- (14) **if** $(r_i \in \text{Elected})$
- (15) $\text{return } ((t \cdot (h + 1) + \text{id} - 1) \cdot \text{rad}, 0)$
- (16) **else**
- (17) $\text{return My Position}$
- (18) **endif**
- (19) **endif**

Fig. 3. Symmetry Breaking

The transformation algorithm is given in Figure 3. It is executed by robots during their Compute phases until one of the two desired configurations is obtained. Its description is based on the polar coordinate system of Section 3.1. The principal idea is to use identifiers of robots in order to break the symmetry of configurations. It does so by making robots choose their destination according to their identity. This way, if two robots with similar views but different identities are activated simultaneously, their views at the end of the cycle will be different and the symmetricity decreases.

Let us observe the following four properties about the algorithm:

1. Denote by c is the center of symmetricity of the initial configuration P_i which is therefore a Weber point (Lemma 8). Since the algorithm makes robots move only through their radius with c (line 15, the Weber point remains invariant during all the execution. This implies that any successive regular/symmetric configuration will have necessarily c as its center of regularity/symmetricity.
2. Again, since robots move only through their radius with c , regularity remains invariant during all the execution. It is thus equal to $\text{reg}(P_i)$. But since $\text{sym}(P_i) = m > 1$, it holds according to Lemma 6 that $\text{reg}(P_i) = \text{sym}(P_i)$. Hence, all the successive configurations will be m -regular, including the final one. This means that if a configuration P' with $\text{sym}(P')$ is reached, it is of type $\mathcal{D}(m)$.
3. At each cycle, the algorithm chooses a set Elected of robots having the same view (equivalence class) (lines 10-13). These Since the algorithm is executed only if the current configuration is m -symmetric ($m > 1$), it holds according to Lemma 2 that $|\text{Elected}| = m$. The positions chosen by robots in Elected are located outside the current SEC. Hence, moving to these positions

cannot increase the symmetricity of the configuration. It follows that the symmetricity of the configurations can only *decrease* during the execution.

4. The actions of robots maintain the same stretch during the whole execution, and it is equal to $F(P_{i+1})$ (line 5).

We prove the following claim about the algorithm:

Lemma 24. *Given the conditions of Lemma 23, if robots are executing algorithm 3, then there exists a time at which they reach a configuration P' with $\text{sym}(P') < \text{SNCD}(m, m')$.*

Proof. Remember that we showed in Item 3 above that symmetricity can only decrease. Assume towards contradiction that it remains always greater or equal to $\text{SNCD}(m, m')$. This means that there exists a time t , a symmetricity $x > \text{SNCD}(m, m')$, such that all the configurations reached after t have symmetricity equal to x . But we assumed that $h > \frac{n}{\text{SNCD}(m, m')}$, which implies that $h > n/x$. Hence, according to Lemma 4, there exists two robots r_1, r_2 with identical views but distinct labels. Let S be the set of robots with the same view as r_1, r_2 . Note that the robots of S form a regular polygon around c , i.e. they lie in the same circle. There exists a time at which the robots of S are elected, i.e. when they become the closer to the center c . Since r_1 and r_2 have distinct labels, they will choose different destinations and the symmetricity will decrease. Contradiction. \square

Definition 16 (m', T). *Let m' be the smallest symmetricity of all the configurations reached by the execution of the algorithm. According to Lemma 24 $m' < \text{SNCD}(m, m')$. Let T be the first reached configuration for which $\text{sym}(T) = m'$. Since all the configurations reached after T if any have a symmetricity equal to m' this means that at each cycle after T is reached, there are m' robots that are elected to move.*

Lemma 25. *If $\text{sym}(T) = m' > 1$, then either T is of type $\mathcal{B}(m')$ or a configuration of type $\mathcal{B}(m')$ can be obtained from T after one cycle.*

Proof. If T is of type $\mathcal{B}(m')$ we are done. If not, then T is of type $\mathcal{C}(m')$ because it is m -symmetric. Hence, the algorithm is executed also by robots when they are at configuration T . Let T^{+1} be the next configuration that is just after T . By definition of m' , T^{+1} must be also m' -symmetric. The external circle of T^{+1} is formed by the m' robots that moved between T and T^{+1} . They necessarily form a regular polygon of m sides, otherwise the symmetricity would have decreased. Moreover, the ratio between the two most external circles is greater than five, hence T^{+1} is of type $\mathcal{B}(m')$. This proves the lemma. \square

Lemma 26. *If $m' = 1$ and all the points in T are collinear, then either T is of type \mathcal{A} or a configuration of type \mathcal{A} can be reached from T by a movement of a single robot.*

Proof. Let T^{-1} denotes the configuration that just precedes T in the execution. Clearly, T^{-1} is also collinear since robots move only through their radius with c . Moreover, $\text{sym}(T^{-1}) = 2 > m' = 1$. We distinguish between two subcases:

1. Only one robot moved between T^{-1} and T . In this case, T is of type \mathcal{A} with appropriate stretch. So the algorithm stops here and the lemma follows.
2. Two robots moved between T^{-1} and T . It can be easily checked that T is not an intermediate configuration (not of type \mathcal{A} , \mathcal{B} , \mathcal{C} or \mathcal{D}). The stretch of T can be however computed by trying to find the center of symmetricity of the precedent configuration T^{-1} . We do this by ignoring the two extreme positions in T (corresponding to the robots that moved between T^{-1} and T). c is the centroid of the remaining positions. Then, having c , we can deduce the stretch using the formula of configurations of type \mathcal{D} . After that, we elect some robot in T and make it move in such a way to form a configuration of type \mathcal{A} with appropriate stretch as showed in Lemma 19.

\square

Lemma 27. *If $m' = 1$ and the points in T are not all collinear, then either T is of type \mathcal{A} or $\mathcal{D}(m)$*

Proof. Let T^{-1} denotes the configuration that just precedes T in the execution. Note that $\text{sym}(T^{-1}) = 2$ since T is by definition the first configuration that reaches symmetricity equal to 1. Let S_1 be the set of robots that moved between T^{-1} and T and let S_2 be the set of robots that remained stationary. Note that $|S_2| \geq n/2$. If $|S_1| = 1$ then the obtained configuration T is of type \mathcal{A} and we are done. Hence we assume that $|S_1| > 1$ and we prove that T is of type \mathcal{D} . For this, it suffices to show that T is neither of type \mathcal{A} nor \mathcal{B} . Let $u = \text{rad}(\text{SEC}(T^{-1}))$. The following two properties hold in T :

- (A1) the distance between any two robots is smaller than $2 \cdot (t+1) \cdot (h+1) \cdot u$. Moreover,
- (A2) $\forall r \in S_2 : \forall r' \in S_1 : |r, r'| \geq (t \cdot (h+1) - 1) \cdot u \geq (t-1) \cdot (h+1) \cdot u$.
- (A3) $\forall r, r' \in S_2 : |r, r'| \leq 2 \cdot u$.

Not \mathcal{B} Assume towards contradiction that T is of type \mathcal{B} . SEC_1 and SEC_2 are defined accordingly. We say $r \in \text{SEC}_1$ if it is on SEC_1 . Note that at least one robot of S_1 must be on SEC_1 , and $\forall r \in T : (r \in \text{SEC}_1) \vee (r \in \text{SEC}_2)$.

Claim (C1). $(\exists r \in S_1 \cap \text{SEC}_2) \Rightarrow (\forall x \in S_2 : x \in \text{SEC}_1)$.

Proof. Fix r to be some robot in $S_1 \cap \text{SEC}_2$. Assume for contradiction that there exists some $x \in S_2 \cap \text{SEC}_2$. By (A2) we conclude that $|x, r| \geq (t-1) \cdot (h+1) \cdot u$. Since both x and r are in SEC_2 , its diameter $\text{diam}(\text{SEC}_2)$ must be greater than $(t-1) \cdot (h+1) \cdot u$. Hence, by definition of pattern \mathcal{B} , $\text{diam}(\text{SEC}_1) > 10 \cdot (t-1) \cdot (h+1) \cdot u$. Thus, we must have at least two robots in SEC_1 distant from each other by more than $5 \cdot (t-1) \cdot (h+1) \cdot u$. When $t > 2$, this contradicts (A1). This proves that $x \notin \text{SEC}_2$. Hence, $\forall x \in S_2 : x \in \text{SEC}_1$. \square

Claim (C2). $\forall r \in S_1 : r \in \text{SEC}_1$

Proof. Assume towards contradiction that some $r \in S_1 \cap \text{SEC}_2$. This implies, according to C1, that $\forall x \in S_2 : x \in \text{SEC}_1$. But $|S_2| \geq n/2$ and there must be at least one robot of S_1 in SEC_1 . Hence this also implies $|\text{SEC}_1| > n/2 + 1$, contradiction because SEC_1 cannot contain the majority of robots in \mathcal{B} . This proves the claim. \square

Claim (C3). $\forall r \in S_2 : r \in \text{SEC}_2$

Proof. Assume that some $r \in S_2 \cap \text{SEC}_1$. Observe that according to C2 all robots x that are in S_1 are on SEC_1 also. Hence, (B1) there exists at least one robot in S_2 that is in SEC_2 , otherwise the latter will be empty. Let this robot be y .

According to (A1), $\text{dist}(x, r) \geq (t-1) \cdot (h+1) \cdot u$. Hence, $\text{radius}(\text{SEC}_1) \geq (t-1) \cdot (h+1) \cdot u$. When $t > 10$ we have $\text{radius}(\text{SEC}_1) \geq 10 \cdot u$. But $\text{radius}(\text{SEC}_1) > 10 \cdot \text{radius}(\text{SEC}_2)$. Hence, (B2) if x_1 is in SEC_1 and x_2 is SEC_2 $|x_1, x_2| \geq 9 \cdot u$.

But we have $r \in S_2 \cap \text{SEC}_1$ and $y \in S_2 \cap \text{SEC}_2$. Since both y and r are in S_2 , we have according to (A3), $\text{dist}(y, r) \leq 2 \cdot u$. This contradicts (B2) and proves the claim. \square

Hence, the robots of SEC_1 are those in S_1 and robots of SEC_2 are those of S_2 . The SEC of S_2 is the SEC of T^{-1} . Hence, the center of SEC_1 is the Weber point. According to the definition of \mathcal{B} , the robots at SEC_1 form a regular polygon. Hence, there is some symmetry maintained between T^{-1} and T . It contradicts the fact that $\text{sym}(T) = 1$. Therefore, T cannot be of type \mathcal{B} .

Not \mathcal{A} can be proved using the same techniques. \square

Proof of Lemma 23 Follows from Lemma 24, Definition 16 and Lemmas 26, 27 and 25.

Theorem 8. Consider a robot network of n robots in configuration P_i . Let $\text{sym}(P_i) = m > 1$ and $\text{sym}(P_{i+1}) = m'$. If $h > \frac{n}{\text{SNCD}(m, m')}$, there exists an algorithm that forms P_{i+1} .

Proof. Follows from Lemmas 21, 22 and 23. \square

The proof of Theorem 7 follows directly from Theorem 8.

7 Special case: Distinct identifiers, Multiplicity points

In this section we consider the case where $l = n$ and patterns may contain *multiplicity points*. In this context, we prove that making the identifiers of robots invisible does not limit their computational power. That is, the series of geometric patterns that can be formed in this case are the same that those we can form when robots are endowed with *visible* distinct identifiers. The following theorem states this result:

Theorem 9. With $n \geq 1$ robots having distinct invisible identifiers, we can for any finite series of distinct patterns $\langle P_1, P_2, \dots, P_m \rangle$ iff for all i , $1 \leq i \leq m$, $\text{size}(P_i) \leq n$ where $\text{size}(P_i)$ is the cardinality of P_i .

Proof. Follows from Lemmas 5.1. and 5.4 in [4] and Lemma 28 below. \square

Lemma 28. given any non-trivial series $\langle P_1, P_2, \dots, P_m \rangle$, where $\text{size}(P_i) > 2$ and any l , $1 \leq l < m$, three robots can form the following series of pattern S :

$$\langle P_1, P_2, \dots, P_l, \text{POINT}, P_{l+1}, P_{l+2}, \dots, P_m, \text{TWO-POINTS} \rangle$$

Proof. To form the series, we need to introduce some intermediate patterns denoted by $C(1, 2; 3)$, $C(1, 3; 2)$, $C(2, 3; 1)$, L_1 , L_3 , S_{123} and S_{213} and whose precise definitions are given below. The obtained series that includes these patterns is the following:

$$\langle P_1, P_2, \dots, P_l, L_1, (1, 2; 3), \text{POINT}, (1, 3; 2), S_{123}, P_{l+1}, P_{l+2}, \dots, P_m, L_3, \text{TWO-POINTS} = (2, 3; 1), S_{213} \rangle$$

The intermediate patterns are defined as follows:

- $C(i, j; l)$, $i, j, l \in \{1, 2, 3\}$ represent a configuration in which robots r_i and r_j are collocated in the same point, and r_l occupies a distinct position.
- L_1 (read "leader r_1 ") and L_3 are two scalene triangles distinct from one another and from any pattern in $\{P_1, P_2, \dots, P_m\}$. Moreover, in L_1 (resp. L_3), the angle whose vertex is r_1 (resp. r_3) is the smallest one.
- S_{123} and S_{213} are two configurations of the same pattern which consists in a scalene triangle with angles $\alpha_1 < \alpha_2 < \alpha_3$. In S_{123} , $\text{angle}(r_i) = \alpha_i$ for any $i \in \{1, 2, 3\}$. In contrast, in S_{213} we have $\text{angle}(r_1) = \alpha_2$ and $\text{angle}(r_2) = \alpha_1$. Note that S_{123} and S_{213} are distinct from the patterns $\{L_1, L_3, P_1, P_2, \dots, P_m\}$.

Now we explain how the transformations between different patterns are handled:

1. $P_i \rightsquigarrow P_{i+1}$: Note that each pattern P_i consists of exactly three distinct points. Hence the transformation between P_i and P_{i+1} is achieved [4] by the movement of only one robot (say r_3).
2. $P_l \rightsquigarrow L_1$: The transformation is achieved by making r_1 moves towards some point such that the obtained pattern is isomorphic to L_1 and r_1 is the vertex of its smallest angle.
3. $L_1 \rightsquigarrow (1, 2; 3)$: Note that r_1 can be distinguished from other robots in L_1 . Hence, the transformation from L_1 to $(1, 2; 3)$ is simply achieved by making r_2 join the location of r_1 .

4. $(1, 2; 3) \rightsquigarrow \text{POINT}$: Here, r_3 has to move towards the multiplicity point.
5. $\text{POINT} \rightsquigarrow (1, 3; 2)$: The latter configuration is obtained by making r_2 move outside the multiplicity point.
6. $(1, 3; 2) \rightsquigarrow S_{123}$ and $(2, 3; 1) \rightsquigarrow S_{213}$: Both transformations are handled by r_3 which cannot distinguish between the two starting configurations $(1, 3; 2)$ and $(2, 3; 1)$. Hence, its actions in both cases are similar. Let p_a denotes the location of the multiplicity point in the starting configuration and let p_b denotes the other location. We make r_3 move to some point p_c such that $\angle(p_c, p_a, p_b) = \alpha_1$, $\angle(p_c, p_b, p_a) = \alpha_2$ and $\angle(p_a, p_c, p_b) = \alpha_3$. It can be easily checked that the obtained configuration is S_{123} in the first case, and S_{213} in the latter case.
7. $S_{123} \rightsquigarrow P_{l+1}$ and $S_{213} \rightsquigarrow P_1$: Note that the starting configurations correspond to the same pattern. Both of the transformations are handled by r_2 . When r_2 observes a configuration consisting of a triangle with angles α_1, α_2 and α_3 , it knows that the current configuration is either S_{123} or S_{213} . It can distinguish between them by looking to its own angle (equal to α_2 in the first case and to α_1 in the second one). Hence, it can execute the appropriate transformation towards P_{l+1} or P_1 .
8. $P_m \rightsquigarrow L_3$: Similar to 2.
9. $L_3 \rightsquigarrow (2, 3; 1)$: Similar to 3.

□

8 Conclusion

In this paper, we considered the problem of formation of series of geometric patterns. We studied the combined effect of obliviousness and anonymity on the computational power of mobile robots with respect to this problem. To this end, we introduced a new model, robots networks with homonyms that encompasses and generalizes the two previously considered models in the literature in which either all robots have distinct identifiers or they are anonymous. Our results suggest that this new model may be a useful tool to get a better insight on how anonymity interacts with others characteristics of the model to limit its power.

Acknowledgement

The authors thank Dr. Shantanu Das for his generous help and helpful comments.

References

1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SODA*, 11(14):1070–1078, 2004.
2. L. Andereg, M. Cieliebak, and G. Prencipe. Efficient algorithms for detecting regular point configurations. *Theoretical Computer Science*, pages 23–35, 2005.
3. R. Chandrasekaran and A. Tamir. Algebraic optimization: the fermat-weber location problem. *Mathematical Programming*, 46(1):219–224, 1990.
4. S. Das, P. Flocchini, N. Santoro, and M. Yamashita. On the computational power of oblivious robots: forming a series of geometric patterns. In *PODC*, pages 267–276. ACM, 2010.
5. C. Delporte-Gallet, H. Fauconnier, R. Guerraoui, A.M. Kermarrec, E. Ruppert, and H. Tran-The. Byzantine agreement with homonyms. *PODC*, 2011.
6. P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Remembering without memory: Tree exploration by asynchronous oblivious robots. *Theoretical Computer Science*, 411(14-15):1583–1598, 2010.
7. B. Katreniak. Biangular circle formation by asynchronous mobile robots. *Structural Information and Communication Complexity*, pages 185–199, 2005.
8. D.E. Knuth, J.H. Morris Jr, and V.R. Pratt. Fast pattern matching in strings. *SIAM journal on computing*, 6:323, 1977.
9. I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal of Computing*, 28(4):1347–1363, 1999.
10. E. WEISZFELD. Sur le point pour lequel la somme des distances de n points donnees est minimum, *tôhoku math. J*, 43:355–386, 1937.